

Amendments to the Claims:

Claims 1, 4, 5, 7, 9, 12, 13, 16, 19, 20, 22, 24, 27, 28, 31, 34, 35, 37, and 38 have been amended. Claims 3, 18, and 33 have been canceled without prejudice. New claims 41-49 have been added. This listing of claims will replace all prior versions, and listings of claims in the application:

Listing of Claims:

- 1 1. (Currently Amended) A computer-implemented method of determining if
2 a query element is included in a set of elements comprising one or more elements from a first
3 domain having a character set of "m" characters, wherein "Z" is the maximum possible length of
4 an element in the first domain and "Y" is the length of the longest element in the set of elements
5 such that $1 \leq Y \leq Z$, the method comprising:
6 building a data structure based upon ~~information identifying elements in the set of~~
7 elements, the data structure comprising a plurality of memory structures headed by a root
8 memory structure, each memory structure in the plurality of memory structures comprising a first
9 memory location and an array of "m" memory locations;
10 receiving information identifying the query element; and
11 using the data structure to determine if the query element is included in the set of
12 elements such that the number of comparisons needed to determine if the query element is
13 included in the set of elements is proportional to a length of the query element and independent
14 of the number of elements in the set of elements.
- 1 2. (Original) The method of claim 1 wherein the query element is of length "q"
2 and at most "q" character comparisons are needed to determine if the query element is included
3 in the set of elements.
- 1 3. (Canceled)
- 1 4. (Currently Amended) The method of claim 3 1 wherein:
2 the data structure comprises a total of (Y+1) levels; and

3 each memory structure in the data structure belongs to a level L , where $(0 \leq L \leq$
4 $Y)$, the level for a particular memory structure denoting the number of memory structures,
5 starting with the root memory structure, that have to be traversed to reach the particular memory
6 structure, the root memory structure belonging to level 0.

1 5. (Currently Amended) The method of claim 4 wherein building the data
2 structure based upon ~~information identifying the elements in~~ the set of elements comprises:
3 for each element "R" in the set of elements, where $R = c_1c_2...c_f$ for some $f \leq Y$,
4 for each c_i where $1 \leq i \leq f$, starting with $i = 1$:
5 (a) selecting a memory structure at level " $(i-1)$ ";
6 (b) if a memory location corresponding to character c_i in the array of
7 memory locations of the presently selected memory structure does not refer to another memory
8 structure in the database, storing an address of a new memory structure at level " i " in the
9 memory location corresponding to character c_i in the array of memory locations of the selected
10 memory structure;
11 (c) selecting the memory structure at level " i " whose address is stored in
12 the memory location corresponding to character c_i in the array of memory locations of the
13 presently selected memory structure;
14 (d) if (" i " is equal to " f "), storing a reference to element "R" in the first
15 memory location of the memory structure selected in step (c);
16 (e) incrementing the value of " i " by one; and
17 (f) repeating steps (b), (c), (d), and (e) for each c_i where (" i " \leq " f ").

1 6. (Original) The method of claim 5 wherein:
2 receiving information identifying the query element comprises:
3 receiving information identifying a query element k , where $k = c_1c_2...c_q$
4 for some $q \leq Z$;
5 using the data structure to determine if the query element is included in the set of
6 elements comprises:

for each c_i of k where $1 \leq i \leq f$, starting with $i = 1$:

(a) selecting a memory structure of the database at level "(i-1)";

(b) if a memory location corresponding to character c_i in the array of memory locations of the presently selected memory structure does not refer to another memory structure in the database, outputting a signal indicating that the query element is not included in the set of elements;

(c) if the memory location corresponding to character c_i in the array of memory locations of the presently selected memory structure stores an address of a memory structure of the database at level "i", selecting the memory structure at level "i" whose address is stored;

(d) incrementing the value of "i" by one; and

(e) repeating steps (b), (c), and (d) while ("i" \leq "q") and the signal indicating that the query element is not included in the set of elements has not been output; and if the signal indicating that the query element is not included in the set of elements has not been output:

determining if the first memory location of the memory structure selected in step (c) refers to the query element; and

if the first memory location of the memory structure selected in step (c) refers to the query element, outputting a signal indicating that the query element is included in the set of elements, else outputting a signal indicating that the query element is not included in the set of elements.

7. (Currently Amended) The method of claim 3 1 wherein building the data structure based upon ~~information identifying the elements in~~ the set of elements comprises:

for each element "R" in the set of elements, where $R = c_1 c_2 \dots c_f$ for some $f \leq Y$, where each character c_i belongs to the character set of the first domain $[[\Sigma]]$, and $1 \leq i \leq f$, storing information in the database indicating the position and identity of each character in element R.

1 8. (Original) The method of claim 7 wherein using the data structure to
2 determine if the query element is included in the set of elements comprises:

3 determining if the query element is included in the set of elements based upon
4 information stored by the database and information identifying characters and their positions in
5 the query element.

1 9. (Currently Amended) The method of claim 3 1 wherein building the data
2 structure based upon ~~information identifying the elements in~~ the set of elements comprises:

3 for each element "R" in the set of elements:

4 (a) selecting the root memory structure of the data structure as the selected
5 memory structure;

6 (b) selecting the first character of element R;

7 (c) if a memory location corresponding to the selected character in the
8 array of memory locations of the selected memory structure does not refer to another memory
9 structure in the data structure, storing an address of a new memory structure in the memory
10 location corresponding to the selected character in the array of memory locations of the presently
11 selected memory structure;

12 (d) selecting the memory structure whose address is stored in the memory
13 location corresponding to the selected character in the array of memory locations of the selected
14 memory structure as the selected memory structure; and

15 (e) if the selected character is the last character of element R, storing a
16 reference to element R in the first memory location of the memory structure selected in step (d),
17 else, selecting the next character of element R, and repeating steps (c), (d),
18 and (e).

1 10. (Original) The method of claim 9 wherein using the data structure to
2 determine if the query element is included in the set of elements comprises:

3 (a) selecting the root memory structure of the data structure as the selected
4 memory structure;

(b) selecting the first character of the query element;

(c) if a memory location corresponding to the selected character in the array of memory locations of the selected memory structure does not refer to another memory structure in the data structure, outputting a signal indicating that the query element is not included in the set of elements,

else, selecting the memory structure whose address is stored as the selected memory element; and

(d) if the selected character is the last character of the query element:

determining if the first memory location of the memory structure selected in step (c) refers to the query element; and

if the first memory location of the memory structure selected in step (c) refers to the query element, outputting a signal indicating that the query element is included in the set of elements, else outputting a signal indicating that the query element is not included in the set of elements;

else:

selecting the next character of the query element , and repeating steps (c) and (d).

11. (Original) The method of claim 1 wherein a size of the data structure is independent of the number of elements in the set of elements.

12. (Currently Amended) A computer-implemented method of determining if a query element is included in a set of elements comprising one or more elements from a first domain having a character set of "m" characters and "Z" is the maximum possible length of an element in the first domain, the method comprising: The method of claim 11 wherein:

~~the set of elements contains elements from a domain Σ having a character set of "m" characters, and wherein "Z" is the maximum possible length of an element in domain Σ ;~~
and

8 building a data structure based upon the set of elements, the data structure
9 ~~comprises~~ comprising "Z" memory structures, each memory structure comprising "m" slots,
10 each slot comprising a first memory location and an array of memory locations, each array of
11 memory locations comprising "(m+1)" memory locations[[]];
12 receiving information identifying the query element; and
13 using the data structure to determine if the query element is included in the set of
14 elements such that the number of comparisons needed to determine if the query element is
15 included in the set of elements is proportional to a length of the query element and independent
16 of the number of elements in the set of elements.

1 13. (Currently Amended) The method of claim 12 wherein building the data
2 structure based upon ~~information identifying the elements in~~ the set of elements comprises:
3 initializing the first memory location and memory locations in the array of
4 memory locations of each slot in each memory structure to null values;
5 for each element "R" in the set of elements, where $R = c_1c_2...c_f$ for some $f \leq Z$,
6 for each c_i where $1 \leq i \leq f$:
7 if ("i" < "f"):
8 storing a non-null value in a memory location corresponding to
9 character c_{i+1} in the array of memory locations of the slot corresponding to c_i of memory
10 structure i ; and
11 if ("i" is equal to "f"):
12 storing a non-null value in the $(m+1)^{th}$ memory location of the
13 array of memory locations of the slot corresponding to c_i of memory structure i ; and
14 storing a reference to element "R" in the first memory location of
15 the slot corresponding to c_i of memory structure i .

1 14. (Original) The method of claim 13 wherein:
2 receiving information identifying the query element comprises:

3 receiving information identifying a query element “k”, where $k = c_1c_2...c_q$

4 for some $q \leq Z$;

5 using the data structure to determine if the query element is included in the set of
6 elements comprises:

7 outputting a signal indicating that the query element is included in the set
8 of elements if, for each c_i of k :

9 if (“i” < “q”), a non-null value is stored in a memory location
10 corresponding to character c_{i+1} in the array of memory locations of the slot corresponding to c_i of
11 memory structure i ; and

12 if (“i” is equal to “q”), a non-null value is stored in the $(m+1)^{th}$
13 memory location in the array of memory locations of the slot corresponding to c_i of memory
14 structure i , and the first memory location of the slot corresponding to c_i of memory structure i
15 refers to the query element.

1 15. (Original) The method of claim 13 wherein:

2 receiving information identifying the query element comprises:

3 receiving information identifying a query element “k”, where $k = c_1c_2...c_q$
4 for some $q \leq Z$; and

5 using the data structure to determine if the query element is included in the set of
6 elements comprises:

7 outputting a signal indicating that the query element is not included in the
8 set of elements if, for any c_i of k :

9 if (“i” is equal to “q”), a null value is stored in the $(m+1)^{th}$ memory
10 location in the array of memory locations of the slot corresponding to c_i of memory structure i , or
11 the first memory location of the slot corresponding to c_i of memory structure i does not refer to
12 the query element; and

13 if ("i" < "q"), a null value is stored in a memory location
14 corresponding to character c_{i+1} in the array of memory locations of the slot corresponding to c_i of
15 memory structure i .

1 16. (Currently Amended) A system for determining if a query element is
2 included in a set of elements comprising one or more elements from a first domain having a
3 character set of "m" characters, wherein "Z" is the maximum possible length of an element in the
4 first domain and "Y" is the length of the longest element in the set of elements such that $1 \leq Y \leq$
5 Z, the system comprising:
6 a processor;
7 a memory coupled to the processor, the memory configured to store a plurality of
8 code modules executable by the processor, the plurality of code modules comprising:
9 a code module for building a data structure based upon ~~information~~
10 ~~identifying elements in the set of elements,~~ the data structure comprising a plurality of memory
11 structures headed by a root memory structure, each memory structure in the plurality of memory
12 structures comprising a first memory location and an array of "m" memory locations;
13 a code module for receiving information identifying the query element;
14 and
15 a code module for using the data structure to determine if the query
16 element is included in the set of elements such that the number of comparisons needed to
17 determine if the query element is included in the set of elements is proportional to a length of the
18 query element and independent of the number of elements in the set of elements.

1 17. (Original) The system of claim 16 wherein the query element is of length "q"
2 and at most "q" character comparisons are needed to determine if the query element is included
3 in the set of elements.

1 18. (Canceled)

1 19. (Currently Amended) The system of claim ~~18~~ 16 wherein:

2 the data structure comprises a total of $(Y+1)$ levels; and
3 each memory structure in the data structure belongs to a level L , where $(0 \leq L \leq$
4 $Y)$, the level for a particular memory structure denoting the number of memory structures,
5 starting with the root memory structure, that have to be traversed to reach the particular memory
6 structure, the root memory structure belonging to level 0.

1 20. (Currently Amended) The system of claim 19 wherein the code module
2 for building the data structure based upon ~~information identifying the elements in the set of~~
3 elements comprises:

4 for each element "R" in the set of elements, where $R = c_1 c_2 \dots c_f$ for some $f \leq Y$,
5 for each c_i where $1 \leq i \leq f$, starting with $i = 1$:

6 (a) a code module for selecting a memory structure at level " $(i-1)$ ";

7 (b) if a memory location corresponding to character c_i in the array of
8 memory locations of the presently selected memory structure does not refer to another memory
9 structure in the database, a code module for storing an address of a new memory structure at
10 level " i " in the memory location corresponding to character c_i in the array of memory locations
11 of the selected memory structure;

12 (c) a code module for selecting the memory structure at level " i " whose
13 address is stored in the memory location corresponding to character c_i in the array of memory
14 locations of the presently selected memory structure;

15 (d) if (" i " is equal to " f "), a code module for storing a reference to element
16 "R" in the first memory location of the memory structure selected in step (c);

17 (e) a code module for incrementing the value of " i " by one; and

18 (f) a code module for repeating steps (b), (c), (d), and (e) for each c_i where
19 (" i " \leq " f ").

1 21. (Original) The system of claim 20 wherein:

2 the code module for receiving information identifying the query element
3 comprises:

4 a code module for receiving information identifying a query element k ,
5 where $k = c_1c_2...c_q$ for some $q \leq Z$;

6 the code module for using the data structure to determine if the query element is
7 included in the set of elements comprises:

8 for each c_i of k where $1 \leq i \leq f$, starting with $i = 1$:

9 (a) a code module for selecting a memory structure of the database at level
10 “(i-1)”;

11 (b) if a memory location corresponding to character c_i in the array of
12 memory locations of the presently selected memory structure does not refer to another memory
13 structure in the database, a code module for outputting a signal indicating that the query element
14 is not included in the set of elements;

15 (c) if the memory location corresponding to character c_i in the array of
16 memory locations of the presently selected memory structure stores an address of a memory
17 structure of the database at level “i”, a code module for selecting the memory structure at level
18 “i” whose address is stored;

19 (d) a code module for incrementing the value of “i” by one; and

20 (e) a code module for repeating steps (b), (c), and (d) while (“i” \leq “q”) and
21 the signal indicating that the query element is not included in the set of elements has not been
22 output; and

23 if the signal indicating that the query element is not included in the set of
24 elements has not been output:

25 a code module for determining if the first memory location of the memory
26 structure selected in step (c) refers to the query element; and

27 if the first memory location of the memory structure selected in step (c)
28 refers to the query element, a code module for outputting a signal indicating that the query
29 element is included in the set of elements, else a code module for outputting a signal indicating
30 that the query element is not included in the set of elements.

1 22. (Currently Amended) The system of claim ~~18~~ 16 wherein the code module
2 for building the data structure based upon ~~information identifying the elements in~~ the set of
3 elements comprises:

4 for each element "R" in the set of elements, where $R = c_1c_2...c_f$ for some $f \leq Y$,
5 where each character c_i belongs to the character set of the first domain $[[\Sigma]]$, and $1 \leq i \leq f$, a code
6 module for storing information in the database indicating the position and identity of each
7 character in element R.

1 23. (Original) The system of claim 22 wherein the code module for using the data
2 structure to determine if the query element is included in the set of elements comprises:

3 a code module for determining if the query element is included in the set of
4 elements based upon information stored by the database and information identifying characters
5 and their positions in the query element.

1 24. (Currently Amended) The system of claim ~~18~~ 16 wherein the code module
2 for building the data structure based upon ~~information identifying the elements in~~ the set of
3 elements comprises:

4 for each element "R" in the set of elements:

5 (a) a code module for selecting the root memory structure of the data
6 structure as the selected memory structure;

7 (b) a code module for selecting the first character of element R;

8 (c) if a memory location corresponding to the selected character in the
9 array of memory locations of the selected memory structure does not refer to another memory
10 structure in the data structure, a code module for storing an address of a new memory structure in
11 the memory location corresponding to the selected character in the array of memory locations of
12 the presently selected memory structure;

13 (d) a code module for selecting the memory structure whose address is
14 stored in the memory location corresponding to the selected character in the array of memory
15 locations of the selected memory structure as the selected memory structure; and

16 (e) if the selected character is the last character of element R, a code
17 module for storing a reference to element R in the first memory location of the memory structure
18 selected in step (d),
19 else, a code module for selecting the next character of element R, and
20 repeating steps (c), (d), and (e)

1 25. (Original) The system of claim 24 wherein the code module for using the data
2 structure to determine if the query element is included in the set of elements comprises:

3 (a) a code module for selecting the root memory structure of the data structure as
4 the selected memory structure;

5 (b) a code module for selecting the first character of the query element;

6 (c) if a memory location corresponding to the selected character in the array of
7 memory locations of the selected memory structure does not refer to another memory structure in
8 the data structure, a code module for outputting a signal indicating that the query element is not
9 included in the set of elements,

10 else, a code module for selecting the memory structure whose address is stored as
11 the selected memory element; and

12 (d) if the selected character is the last character of the query element:

13 a code module for determining if the first memory location of the memory
14 structure selected in step (c) refers to the query element; and

15 if the first memory location of the memory structure selected in step (c)
16 refers to the query element, a code module for outputting a signal indicating that the query
17 element is included in the set of elements, else a code module for outputting a signal indicating
18 that the query element is not included in the set of elements;

19 else:

20 a code module for selecting the next character of the query element , and
21 repeating steps (c) and (d).

1 26. (Original) The system of claim 16 wherein a size of the data structure is
2 independent of the number of elements in the set of elements.

1 27. (Currently Amended) A system for determining if a query element is
2 included in a set of elements comprising one or more elements from a first domain having a
3 character set of "m" characters, and "Z" is the maximum possible length of an element in the
4 first domain, the system comprising: ~~The system of claim 26 wherein:~~
5 ~~the set of elements contains elements from a domain Σ having a character set of~~
6 ~~"m" characters, and wherein "Z" is the maximum possible length of an element in domain Σ ;~~
7 and
8 a processor; and
9 a memory coupled to the processor, the memory configured to store a plurality of
10 code modules executable by the processor, the plurality of code modules comprising:
11 a code module for building a data structure based upon the set of elements,
12 the data structure ~~comprises~~ comprising "Z" memory structures, each memory structure
13 comprising "m" slots, each slot comprising a first memory location and an array of memory
14 locations, each array of memory locations comprising "(m+1)" memory locations[[]];
15 a code module for receiving information identifying the query element;
16 and
17 a code module for using the data structure to determine if the query
18 element is included in the set of elements such that the number of comparisons needed to
19 determine if the query element is included in the set of elements is proportional to a length of the
20 query element and independent of the number of elements in the set of elements.

1 28. (Currently Amended) The system of claim 27 wherein the code module
2 for building the data structure based upon ~~information identifying the elements in the set of~~
3 ~~elements comprises:~~
4 a code module for initializing the first memory location and memory locations in
5 the array of memory locations of each slot in each memory structure to null values;
6 for each element "R" in the set of elements, where $R = c_1c_2...c_f$ for some $f \leq Z$,
7 for each c_i where $1 \leq i \leq f$:

8 if (“i” < “f”):
9 a code module for storing a non-null value in a memory location
10 corresponding to character c_{i+1} in the array of memory locations of the slot corresponding to c_i of
11 memory structure i ; and
12 if (“i” is equal to “f”):
13 a code module for storing a non-null value in the $(m+1)^{th}$ memory
14 location of the array of memory locations of the slot corresponding to c_i of memory structure i ;
15 and
16 a code module for storing a reference to element “R” in the first
17 memory location of the slot corresponding to c_i of memory structure i .

1 29. (Original) The system of claim 28 wherein:
2 the code module for receiving information identifying the query element
3 comprises:
4 a code module for receiving information identifying a query element “k”,
5 where $k = c_1c_2...c_q$ for some $q \leq Z$;
6 the code module for using the data structure to determine if the query element is
7 included in the set of elements comprises:
8 a code module for outputting a signal indicating that the query element is
9 included in the set of elements if, for each c_i of k :
10 if (“i” < “q”), a non-null value is stored in a memory location
11 corresponding to character c_{i+1} in the array of memory locations of the slot corresponding to c_i of
12 memory structure i ; and
13 if (“i” is equal to “q”), a non-null value is stored in the $(m+1)^{th}$
14 memory location in the array of memory locations of the slot corresponding to c_i of memory
15 structure i , and the first memory location of the slot corresponding to c_i of memory structure i
16 refers to the query element.

1 30. (Original) The system of claim 28 wherein:

2 the code module for receiving information identifying the query element
3 comprises:
4 a code module for receiving information identifying a query element "k",
5 where $k = c_1 c_2 \dots c_q$ for some $q \leq Z$; and
6 the code module for using the data structure to determine if the query element is
7 included in the set of elements comprises:
8 a code module for outputting a signal indicating that the query element is
9 not included in the set of elements if, for any c_i of k :
10 if ("i" is equal to "q"), a null value is stored in the $(m+1)^{th}$ memory
11 location in the array of memory locations of the slot corresponding to c_i of memory structure i , or
12 the first memory location of the slot corresponding to c_i of memory structure i does not refer to
13 the query element; and
14 if ("i" < "q"), a null value is stored in a memory location
15 corresponding to character c_{i+1} in the array of memory locations of the slot corresponding to c_i of
16 memory structure i .

1 31. (Currently Amended) A computer program product stored on a computer-
2 readable storage medium for determining if a query element is included in a set of elements
3 comprising one or more elements from a first domain having a character set of "m" characters,
4 wherein "Z" is the maximum possible length of an element in the first domain and "Y" is the
5 length of the longest element in the set of elements such that $1 \leq Y \leq Z$, the computer program
6 product comprising:
7 code for building a data structure based upon ~~information identifying elements in~~
8 the set of elements, the data structure comprising a plurality of memory structures headed by a
9 root memory structure, each memory structure in the plurality of memory structures comprising a
10 first memory location and an array of "m" memory locations;
11 code for receiving information identifying the query element; and
12 code for using the data structure to determine if the query element is included in
13 the set of elements such that the number of comparisons needed to determine if the query

14 element is included in the set of elements is proportional to a length of the query element and
15 independent of the number of elements in the set of elements.

1 32. (Original) The computer program product of claim 31 wherein the query
2 element is of length "q" and at most "q" character comparisons are needed to determine if the
3 query element is included in the set of elements.

1 33. (Canceled)

1 34. (Currently Amended) The computer program product of claim ~~33~~ 31
2 wherein:
3 the data structure comprises a total of (Y+1) levels; and
4 each memory structure in the data structure belongs to a level L, where ($0 \leq L \leq$
5 Y), the level for a particular memory structure denoting the number of memory structures,
6 starting with the root memory structure, that have to be traversed to reach the particular memory
7 structure, the root memory structure belonging to level 0.

1 35. (Currently Amended) The computer program product of claim ~~33~~ 31
2 wherein the code for building the data structure based upon ~~information identifying the elements~~
3 ~~in~~ the set of elements comprises:
4 for each element "R" in the set of elements, where $R = c_1 c_2 \dots c_f$ for some $f \leq Y$,
5 where each character c_i belongs to the character set of the first domain $[[\Sigma]]$, and $1 \leq i \leq f$, code
6 for storing information in the database indicating the position and identity of each character in
7 element R.

1 36. (Original) The computer program product of claim 35 wherein the code for
2 using the data structure to determine if the query element is included in the set of elements
3 comprises:

code for determining if the query element is included in the set of elements based upon information stored by the database and information identifying characters and their positions in the query element.

37. (Currently Amended) A computer program product stored on a computer-readable storage medium for determining if a query element is included in a set of elements comprising one or more elements from a first domain having a character set of "m" characters, and "Z" is the maximum possible length of an element in the first domain, the computer program product comprising: ~~The computer program product of claim 31 wherein:~~

~~a size of the data structure is independent of the number of elements in the set of elements;~~

~~the set of elements contains elements from a domain Σ having a character set of "m" characters, and wherein "Z" is the maximum possible length of an element in domain Σ ;~~
and

code for building a data structure based upon the set of elements, the data structure comprises comprising "Z" memory structures, each memory structure comprising "m" slots, each slot comprising a first memory location and an array of memory locations, each array of memory locations comprising "(m+1)" memory locations[.];

code for receiving information identifying the query element; and
code for using the data structure to determine if the query element is included in the set of elements such that the number of comparisons needed to determine if the query element is included in the set of elements is proportional to a length of the query element and independent of the number of elements in the set of elements.

38. (Currently Amended) The computer program product of claim 37 wherein the code for building the data structure based upon ~~information identifying the elements in the~~ set of elements comprises:

code for initializing the first memory location and memory locations in the array of memory locations of each slot in each memory structure to null values;

6 for each element "R" in the set of elements, where $R = c_1c_2...c_f$ for some $f \leq Z$,
7 for each c_i where $1 \leq i \leq f$:
8 if ("i" < "f"):
9 code for storing a non-null value in a memory location
10 corresponding to character c_{i+1} in the array of memory locations of the slot corresponding to c_i of
11 memory structure i ; and
12 if ("i" is equal to "f"):
13 code for storing a non-null value in the $(m+1)^{th}$ memory location
14 of the array of memory locations of the slot corresponding to c_i of memory structure i ; and
15 code for storing a reference to element "R" in the first memory
16 location of the slot corresponding to c_i of memory structure i .

1 39. (Original) The computer program product of claim 38 wherein:
2 the code for receiving information identifying the query element comprises:
3 code for receiving information identifying a query element "k", where
4 $k = c_1c_2...c_q$ for some $q \leq Z$;
5 the code for using the data structure to determine if the query element is included
6 in the set of elements comprises:
7 code for outputting a signal indicating that the query element is included
8 in the set of elements if, for each c_i of k :
9 if ("i" < "q"), a non-null value is stored in a memory location
10 corresponding to character c_{i+1} in the array of memory locations of the slot corresponding to c_i of
11 memory structure i ; and
12 if ("i" is equal to "q"), a non-null value is stored in the $(m+1)^{th}$
13 memory location in the array of memory locations of the slot corresponding to c_i of memory
14 structure i , and the first memory location of the slot corresponding to c_i of memory structure i
15 refers to the query element.

1 40. (Original) The computer program product of claim 38 wherein:

2 the code for receiving information identifying the query element comprises:
3 code for receiving information identifying a query element "k", where
4 $k = c_1c_2...c_q$ for some $q \leq Z$;
5 the code for using the data structure to determine if the query element is included
6 in the set of elements comprises:
7 code for outputting a signal indicating that the query element is not
8 included in the set of elements if, for any c_i of k :
9 if ("i" is equal to "q"), a null value is stored in the $(m+1)^{th}$ memory
10 location in the array of memory locations of the slot corresponding to c_i of memory structure i , or
11 the first memory location of the slot corresponding to c_i of memory structure i does not refer to
12 the query element; and
13 if ("i" < "q"), a null value is stored in a memory location
14 corresponding to character c_{i+1} in the array of memory locations of the slot corresponding to c_i of
15 memory structure i .

1 41. (New) A computer-implemented method of determining if a query
2 element is included in a set of elements comprising one or more elements from a first domain
3 having a character set of "m" characters, wherein "D" is the total number of characters in all the
4 elements in the set of elements and "H" is the number of characters in the query element, the
5 method comprising:

6 building a data structure based upon the set of elements, wherein the size of the
7 data structure is $O(D)$;

8 receiving information identifying the query element; and

9 using the data structure to determine if the query element is included in the set of
10 elements such that the number of comparisons needed to determine if the query element is
11 included in the set of elements is proportional to $(H \log m)$.

1 42. (New) The method of claim 41 wherein building the data structure
2 comprises building the data structure comprising a plurality of memory structures headed by a

3 root memory structure, each memory structure in the plurality of memory structures comprising a
4 first memory location and a balanced tree.

1 43. (New) A system for determining if a query element is included in a set of
2 elements comprising one or more elements from a first domain having a character set of "m"
3 characters, wherein "D" is the total number of characters in all the elements in the set of
4 elements, and "H" is the number of characters in the query element, the system comprising:
5 a processor; and
6 a memory coupled to the processor, the memory configured to store a plurality of
7 code modules executable by the processor, the plurality of code modules comprising:
8 a code module for building a data structure based upon the set of elements,
9 wherein the size of the data structure is $O(D)$;
10 a code module for receiving information identifying the query element;
11 and
12 a code module for using the data structure to determine if the query
13 element is included in the set of elements such that the number of comparisons needed to
14 determine if the query element is included in the set of elements is proportional to $(H \log m)$.

1 44. (New) The system of claim 43 wherein the code module for building the
2 data structure comprises a code module for building the data structure comprising a plurality of
3 memory structures headed by a root memory structure, each memory structure in the plurality of
4 memory structures comprising a first memory location and a balanced tree.

1 45. (New) A computer program product stored on a computer-readable
2 storage medium for determining if a query element is included in a set of elements comprising
3 one or more elements from a first domain having a character set of "m" characters, wherein "D"
4 is the total number of characters in all the elements in the set of elements, and "H" is the number
5 of characters in the query element, the computer program product comprising:
6 code for building a data structure based upon the set of elements, wherein the size
7 of the data structure is $O(D)$;

8 code for receiving information identifying the query element; and
9 code for using the data structure to determine if the query element is included in
10 the set of elements such that the number of comparisons needed to determine if the query
11 element is included in the set of elements is proportional to $(H \log m)$.

1 46. (New) The computer program product of claim 45 wherein the code for
2 building the data structure comprises code for building the data structure comprising a plurality
3 of memory structures headed by a root memory structure, each memory structure in the plurality
4 of memory structures comprising a first memory location and a balanced tree.

1 47. (New) A computer-implemented method of determining if a query
2 element is included in a set of elements comprising one or more elements from a first domain
3 having a character set of "m" characters, and "Z" is the maximum possible length of an element
4 in the first domain, the method comprising:

5 building a data structure based upon the set of elements, wherein the size of the
6 data structure is $O(Z * m^2)$;

7 receiving information identifying the query element; and
8 using the data structure to determine if the query element is included in the set of
9 elements.

1 48. (New) A system for determining if a query element is included in a set of
2 elements comprising one or more elements from a first domain having a character set of "m"
3 characters, and "Z" is the maximum possible length of an element in the first domain, the system
4 comprising:

5 a processor; and
6 a memory coupled to the processor, the memory configured to store a plurality of
7 code modules executable by the processor, the plurality of code modules comprising:
8 a code module for building a data structure based upon the set of elements,
9 wherein the size of the data structure is $O(Z * m^2)$;

10 a code module for receiving information identifying the query element;
11 and
12 a code module for using the data structure to determine if the query
13 element is included in the set of elements.

1 49. (New) A computer program product stored on a computer-readable
2 storage medium for determining if a query element is included in a set of elements comprising
3 one or more elements from a first domain having a character set of "m" characters, and "Z" is the
4 maximum possible length of an element in the first domain, the computer program product
5 comprising:
6 code for building a data structure based upon the set of elements, wherein the size
7 of the data structure is $O(Z * m^2)$;
8 code for receiving information identifying the query element; and
9 code for using the data structure to determine if the query element is included in
10 the set of elements.